Data Cleaning and Analysis with Open Refine

Workshop data folder link: go.ncsu.edu/refine

Open Refine

Open Refine (formerly Google Refine) is a data manipulation and cleaning tool that runs in your browser. The data is not stored online.

Open Refine allows you to clean data and it saves all of your edits as steps, which you can Undo and Redo (unlike in Excel)

If you know how to use regular expressions, Open Refine is even more powerful.

GREL (*OpenRefine Expression Language* *(GREL)* is a programming language for Open Refine that allows you to do many types of manipulations of the data. We will use it in this workshop.

To open the software:
Press command + Space Bar: type in "Refine"
Or, search for it in the Applications folder

Download this data set from the data folder:
NC_LEED_2002-2010.csv

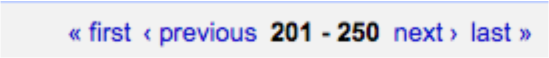
In Refine, click "Create Project"

Choose Files > Open > Next

Click **Create Project**

Click on Show 50 Rows to see more data

You can see your data by clicking on **first/previous/next/last** options in the menu:



« first ‹ previous **201 - 250** next › last »

Scanning over the data we can see that it is pretty clean, but there are some things that need to be changed.

1. Fix inconsistencies in upper/lower case

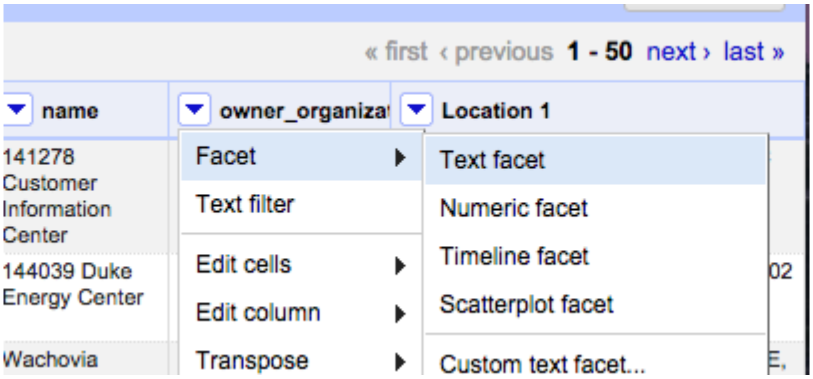
Click on the down arrow in the **name** column

Select Edit Cells > Common Transforms > To titlecase

This should make all values in the cells have a capitalized first letter of each word

2. Cluster and edit column owner_organization

owner_organization > Facet > Text Facet

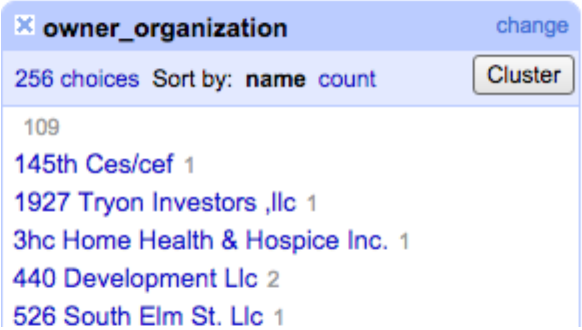


141278 Customer Information Center

144039 Duke Energy Center

Wachovia

in the left column, click the Cluster button



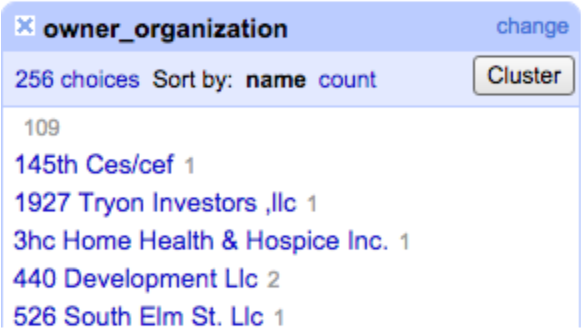
	109
145th Ces/cef	1
1927 Tryon Investors ,llc	1
3hc Home Health & Hospice Inc.	1
440 Development Llc	2
526 South Elm St. Llc	1

Here you should be able to edit any inconsistencies in organization names.

Choose what you would like to Merge then Select **"Merge Selected & Re-Cluster"**

Select different options of clustering algorithms from the "Key Function" dropdown menu to find different potential clusters.

3. Edit directly in the cluster menu



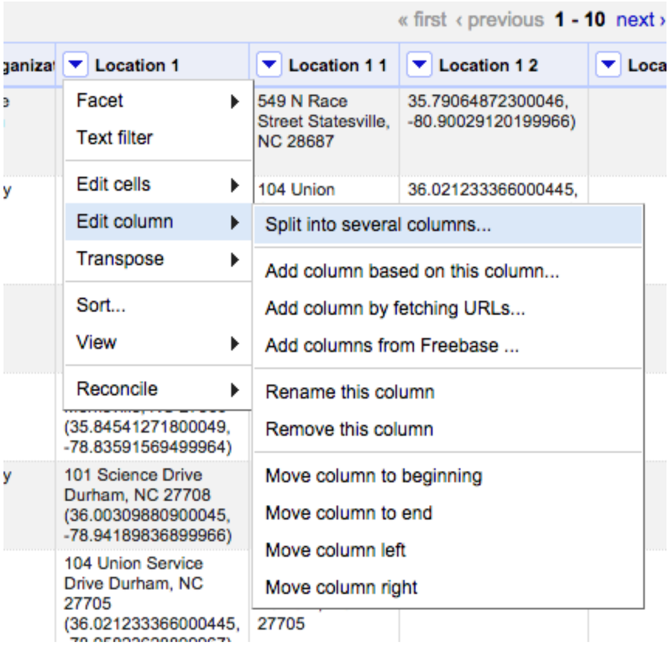
256 choices Sort by: name count

109
145th Ces/cef 1
1927 Tryon Investors ,llc 1
3hc Home Health & Hospice Inc. 1
440 Development Llc 2
526 South Elm St. Llc 1

Here you can scroll through and manually edit based on any errors you find. Click on edit to the right of the name of the item.

4. Separate address from Lat/Longitude

In Location 1, choose Edit column > Split into several columns...



Type in the open parenthesis sign "(" as the delimiter value (without quote marks)

This should split the column so that the address is in its own column.

5. Split the column that has latitude and longitude values:

Edit column > Split into several columns…

“,” as the delimiter

rename latitude column (Edit column > Rename this column)

6. Remove trailing parenthesis) from longitude column

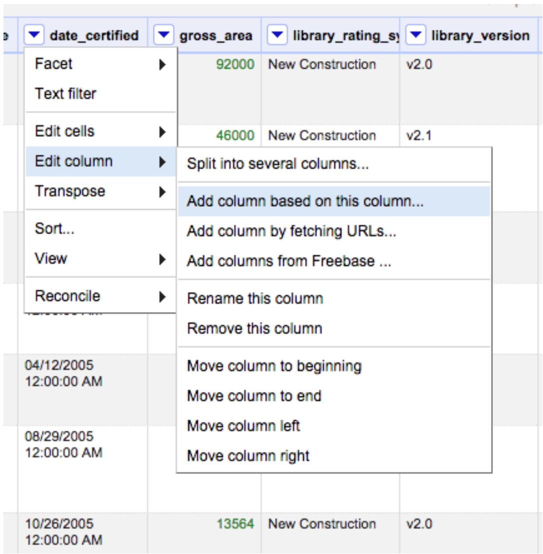
Edit cells > transform

In the expression box, type or copy/paste the following:

```
value.replace(")", "")
```

To make the longitude values into numbers (they are currently strings) type:

Edit cells…
Common transforms > To number



7. Create a new column for the year only.

date_certified column, click on the down arrow

choose **Edit column…**

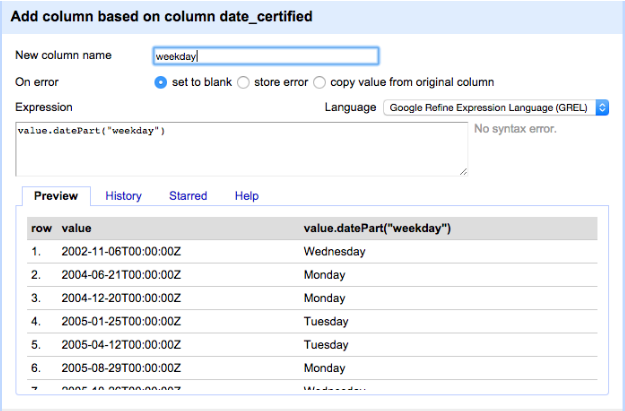
then **“Add column based on this column…”**

Type in a new column name

In the Expression window that pops up, type the following formula:

```
value.toDate('MM/dd/yy').toString('yyyy')
```

This will convert the date that you have from its current format (MM/dd/yy) to the the 4-character year (‘yyyy’)



Add column based on column date_certified

New column name: weekday

On error: ● set to blank ○ store error ○ copy value from original column

Expression — Language: Google Refine Expression Language (GREL)

```
value.datePart("weekday")
```

No syntax error.

row	value	value.datePart("weekday")
1.	2002-11-06T00:00:00Z	Wednesday
2.	2004-06-21T00:00:00Z	Monday
3.	2004-12-20T00:00:00Z	Monday
4.	2005-01-25T00:00:00Z	Tuesday
5.	2005-04-12T00:00:00Z	Tuesday
6.	2005-08-29T00:00:00Z	Monday

8. Add a weekday field based on the date_certified colum

choose **Edit column…**

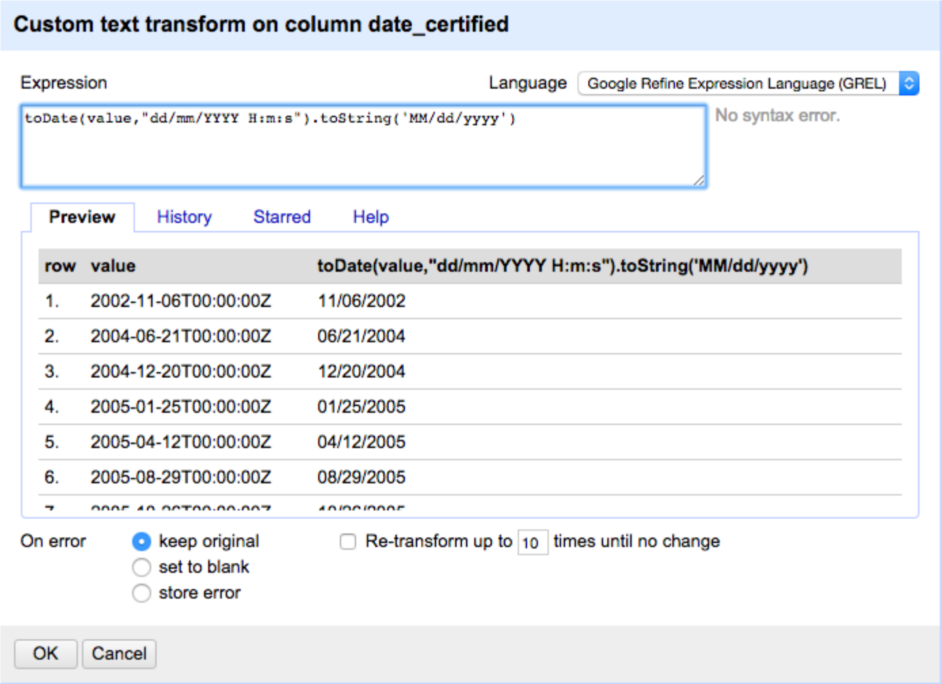
then **“Add column based on this column…”**

```
value.datePart(“weekday”)
```

9. Change date_certified column to mm/dd/yyyy format

```
toDate(value,"dd/mm/YYYY H:m:s").toString('MM/dd/yyyy')
```

Using Extract/Apply to Clean Future Datasets



Custom text transform on column date_certified

Expression:

```
toDate(value,"dd/mm/YYYY H:m:s").toString('MM/dd/yyyy')
```

Language: Google Refine Expression Language (GREL)

No syntax error.

row	value	toDate(value,"dd/mm/YYYY H:m:s").toString('MM/dd/yyyy')
1.	2002-11-06T00:00:00Z	11/06/2002
2.	2004-06-21T00:00:00Z	06/21/2004
3.	2004-12-20T00:00:00Z	12/20/2004
4.	2005-01-25T00:00:00Z	01/25/2005
5.	2005-04-12T00:00:00Z	04/12/2005
6.	2005-08-29T00:00:00Z	08/29/2005
7.	2005-10-26T00:00:00Z	10/26/2005

On error ● keep original ○ set to blank ○ store error

□ Re-transform up to 10 times until no change

With Open Refine, you can save your work and apply those changes to future data sets!

Here’s how:

On the left hand column, click Undo/Redo

Click the Extract button

Make sure all values are selected (they should be highlighted blue). command +A to select all

Copy with command + c

Open Text Edit on your Mac (or any text editor). Paste the JSON code in there

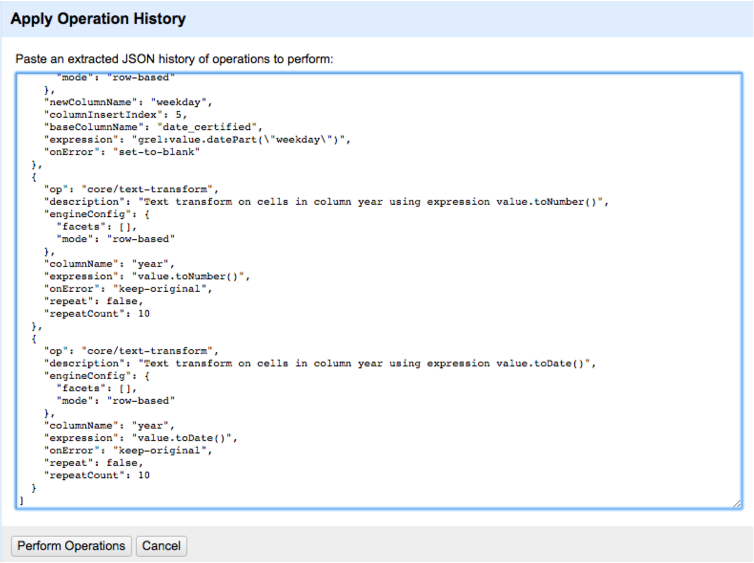
Now load **NC_LEED_2011 2014.xslx** data set into Refine
Open > Create Project > Choose files > NC_LEED_2011 2014.xslx > Next> Create project

This data set is structured the same as the previous one, only different years.

Click Undo/Redo

Click Apply

Paste the JSON code into the window and click **Perform Operations.**



Apply Operation History

Paste an extracted JSON history of operations to perform:

```
      "mode": "row-based"
    },
    "newColumnName": "weekday",
    "columnInsertIndex": 5,
    "baseColumnName": "date_certified",
    "expression": "grel:value.datePart(\"weekday\")",
    "onError": "set-to-blank"
  },
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column year using expression value.toNumber()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "year",
    "expression": "value.toNumber()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
  },
  {
    "op": "core/text-transform",
    "description": "Text transform on cells in column year using expression value.toDate()",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "year",
    "expression": "value.toDate()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10
  }
]
```

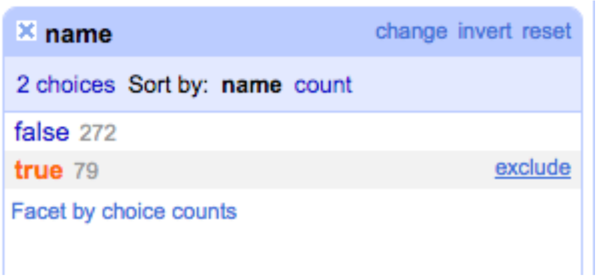
10. Find the blanks in **name** column

In name column, select Facet > Customized facets > Facet by blank

This will give you a facet window on the left-side of the screen with “true” and “false” groups.

The “true” group are all of the rows with a blank value for this column. Click on true to find all of the blanks

To exclude the blanks, click exclude



false 272
true 79

11. Subset the data by creating a numeric facet on the gross_area column

This gives you a slider where you can filter by range of values to see a subset of the data. Move the slider in the facet window to select a subset.

Gross_area column dropdown arrow > Facet > Numeric facet

12. Export the data

Click Export and choose your preferred file type.

13. Going farther with Open Refine: Data Augmentation

There is much more that you can do with Open Refine, for example, to bring data in from the web in various formats.

For example: Finding lat/longitude data from the web, given an address: (this can take awhile depending on the size of the dataset)

Find the column with the street add (Location 1) column in the LEED data set, click the dropdown arrow

Choose **Add column by fetching URLs…**
Type this into the expression window:

"http://nominatim.openstreetmap.org/search?format=json&app=google-refine&q=" + escape(value, "url")

This column will retrieve a lot of data in json format that then needs to be parsed.

See this video for more information: https://www.youtube.com/watch?v=5tsyz3ibYzk

Test Yourself!

Now try doing some data cleaning in OpenRefine with the “**practice data.xslx**” file in the materials folder.